**User’s Manual**

*Social Network Analysis and Simulation Tool*

Xinyue Ye, New Jersey Institute of Technology

Ming-Hsiang Tsou, San Diego State University

Developer and Contributor Team (by Alphabetical Order):

Lu Chen, Zhuo Chen, Lanxue Dang, Zhenliang Feng, Sagar Jha, Jay Lee, Bozhi Wang, Wuji Liu

Updated on Feb 5, 2019

Code is at:

https://github.com/socialnetworktool/SocialNetworkSimulator

**USER’S MANUAL**

**TABLE OF CONTENTS**

1 GENERAL INFORMAITON

1.1 System Overview

1.2 Organization of the Manual

1.3 Acronyms and Abbreviations

2 GETTING STARTED

2.1 Environment Requirements

2.2 Installation

2.3 Start

2.4 System Menu

3 NETWORK GENERATORS

3.1 Generate a Single Simulated Network

3.2 Generate a Complex Simulated Network

3.3 Save a Network

3.4 Load a Network

4 NETWORK ANALYSES

5 COMMUNITY DETECTION

6 INFORMAITON DIFFUSION SIMULATION (USER LEVEL)

7 INFORMAITON DIFFUSION SIMULATION (CITY LEVEL)

8 DATA PROCESSING

**1 GENERAL INFORMAITON**

1.1 System Overview

Currently, social media has been playing an important role in the process of information diffusion. Exploring the pattern of message propagation on social network help us better prepare for natural disasters or human crises. So, we developed models, algorithms, and tools to generate simulated networks, analyze simulated networks, and simulate information diffusion on social network over time.

This software provides the following functions:

(1) Simulated social network generators. This network generator is designed to generate networks with different configuration, such as Random network, Full network, Circle Network, Prefattached network, SmallWord network, and so on.

(2) Network analysis. Network analysis was designed to explore the characteristics of a network. Now this module can be used to analysis characters of network and nodes of network.

(3) Community detection. In this tool, we integrated two algorithms for community detection, including CNM (Clauset-Newman-Moore community detection) and GN (Girvan-Newman community detection).

(4) Simulating information diffusion on simulated network. In this module, we implemented two information diffusion models, independent cascade model (IC model) and linear threshold model (LT model).

(5) Case study. This module provides functions to conduct case study.

(6) Data preprocessing. Because there are some functions may need you load your data that organized as a special format. This module help user prepare data for further use.

1.2 Organization of the Manual

Section 1 is introduction of this tool and this manual. Section 2 provide a guide for user to install this software and have an overlook at it. From Section 3, each section presents a detailed guideline for the corresponding module.

**2 GETTING STARTED**

2.1 Environment Requirements2.2.1 Installation(Choice 1)

Before the code is being used some installation is required so errors can be avoided. All the packages locate in the folder named softwares\_and\_packages.

（1) Python 2.7 x64 version is required.

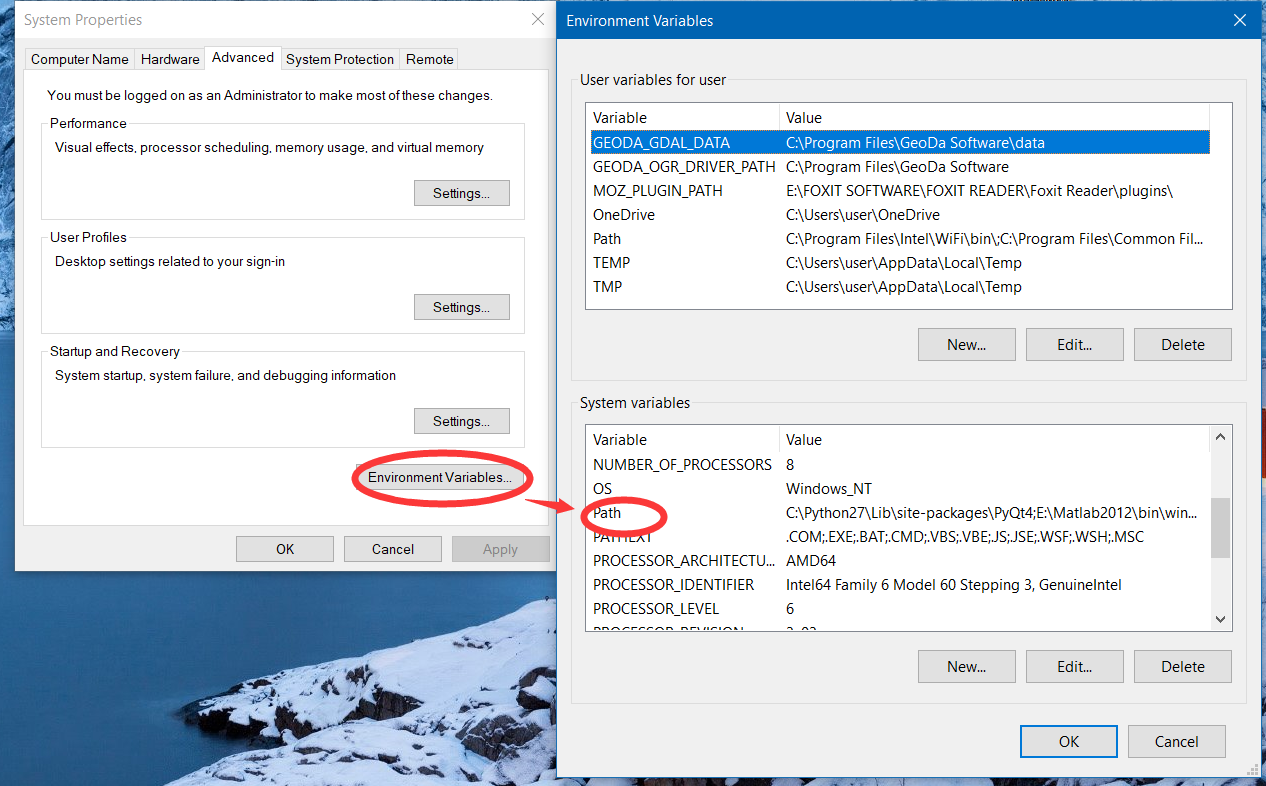
a. Double click *python-2.7.13.amd64.msi* to install python 2.7 64 version.

b. Follow steps below to set the path variables in the environment variables.

1) Right-click **This PC**, and then click **Properties**.

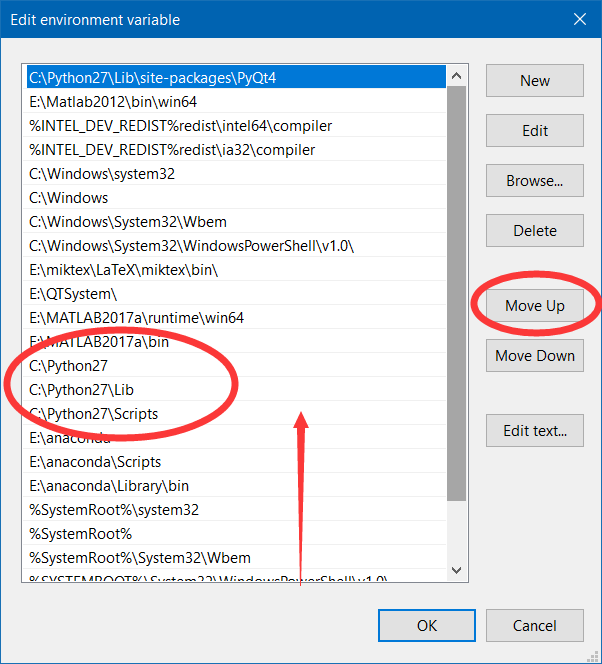
2) Click **Advance system setting**.

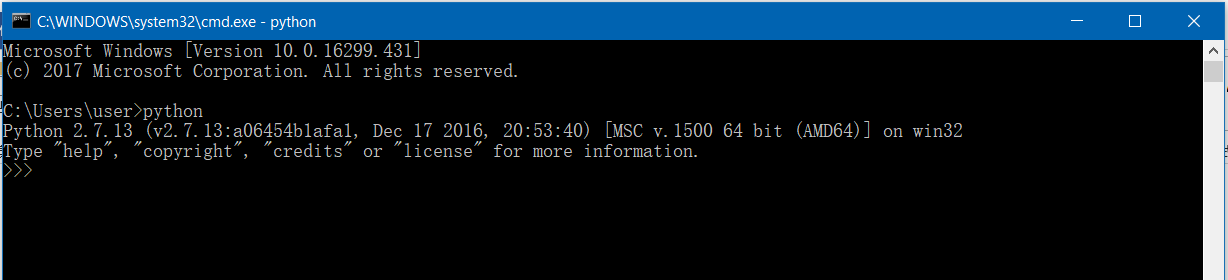
3) Click **Environment variables**.

 4) Go to the system variables and change the **Path** variable.

5) If you install python at C:\Python27, add the following paths to **Path** variable. Otherwise you need to change the path according your actual path.

i. For Windows 7 users, you should add “C:\Python27\”; “C:\Python27\Lib\”; “C:\Python27\Scripts\” connected with symbol “;”.ii. For Windows 8\Windows 10 users, you can double click “path”, then click “New” to add these three paths into this list.

iii. If you have installed python3.x or anaconda, you need to use “Move Up” button to ensure them in front of these programs. You can use command “python” by command prompt to check your version.



This gives the user the access to use all the scripts available in python from command prompt. It will let the user fire the pip command from the command prompt.

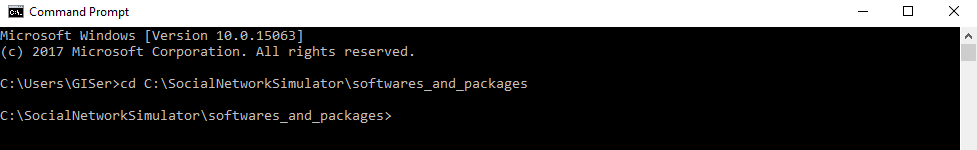
（2) Double click *PyQt4-4.11.4-gpl-Py2.7-Qt4.8.7-x64.exe(softwares\_and\_packages lack this software)* to install PyQt package.

（3) Double click *vcredist\_x64.exe* to install it.

（4) Install module numpy, matplotlib, Snap ,xlwt and xlrd.

a. Open command prompt and change path to the location where packages are. Here you are supposed to extract the tool file to **C:\SocialNetworkSimulator.** Use the command:

**cd C:\SocialNetworkSimulator\softwares\_and\_packages** to change the path.



b. Execute the following commands using command prompt.

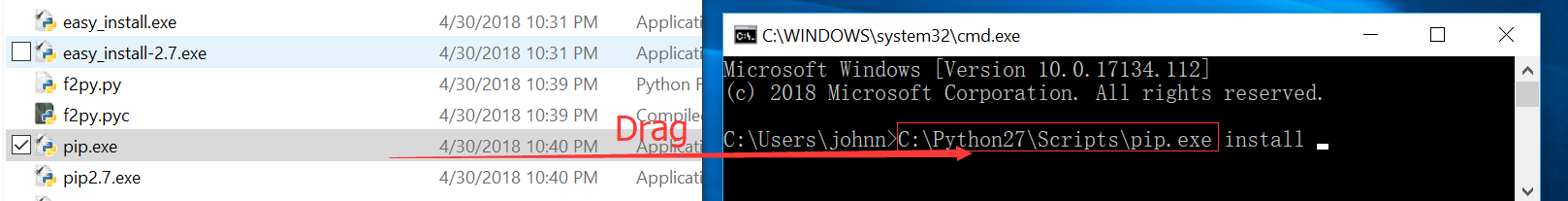
**pip install numpy-1.13.1-cp27-none-win\_amd64.whl**

**pip install matplotlib-2.0.2-cp27-cp27m-win\_amd64.whl**

**pip install snap-4.0.0-4.0-Win-x64-py2.7.zip**

**pip install xlrd\_with\_formulas-1.0.0-py2.py3-none-any.whl**

* Possible Error: [***'pip' is not recognized as an internal or external command***](https://stackoverflow.com/questions/23708898/pip-is-not-recognized-as-an-internal-or-external-command) *”*
* Solution: If users are unable to call pip command in the console. Please go to the directory where pip.exe is located. For example, **"C:\Python27\Scripts\pip2.exe". Drag “pip.exe” into the console** and add “**install**” followed by the **file** that need to be installed. User can also drag the file instead of manually type it.



2.2.2 Installation (Choice 2)

Before taking any further actions, make sure that you have installed Anaconda

(1). Create Python Environment with Anaconda:

* Execute the following command:  
  **conda create --name python4SNS python=2.7**

**conda activate python4SNS**

(2). Install PyQT4:

1. Download whl file from : [https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyqt4](https://www.lfd.uci.edu/~gohlke/pythonlibs/)
2. Execute the following command:  
   **pip install PyQt4-4.11.4-cp27-cp27m-win\_amd64.whl  
   ((For Mac users: conda install pyqt=4)**

(3) Install SNAP and other libs:

1. Extract zipped SNAP file to your local file system.
2. Execute the following command to install:

**cd snap-4.0.0-4.0-Win-x64-py2.7**

**python setup.py install.**

**pip install xlwt xlrd numpy matplotlib prettytable**

2.3 Start

To start this software, you can double click “SocialNetworkSimulator.py” from the document of “C:\ SocialNetworkSimulator”. 2.4 System Menu

After starting the software tool, you can see there is a menu bar on the top of the main window.



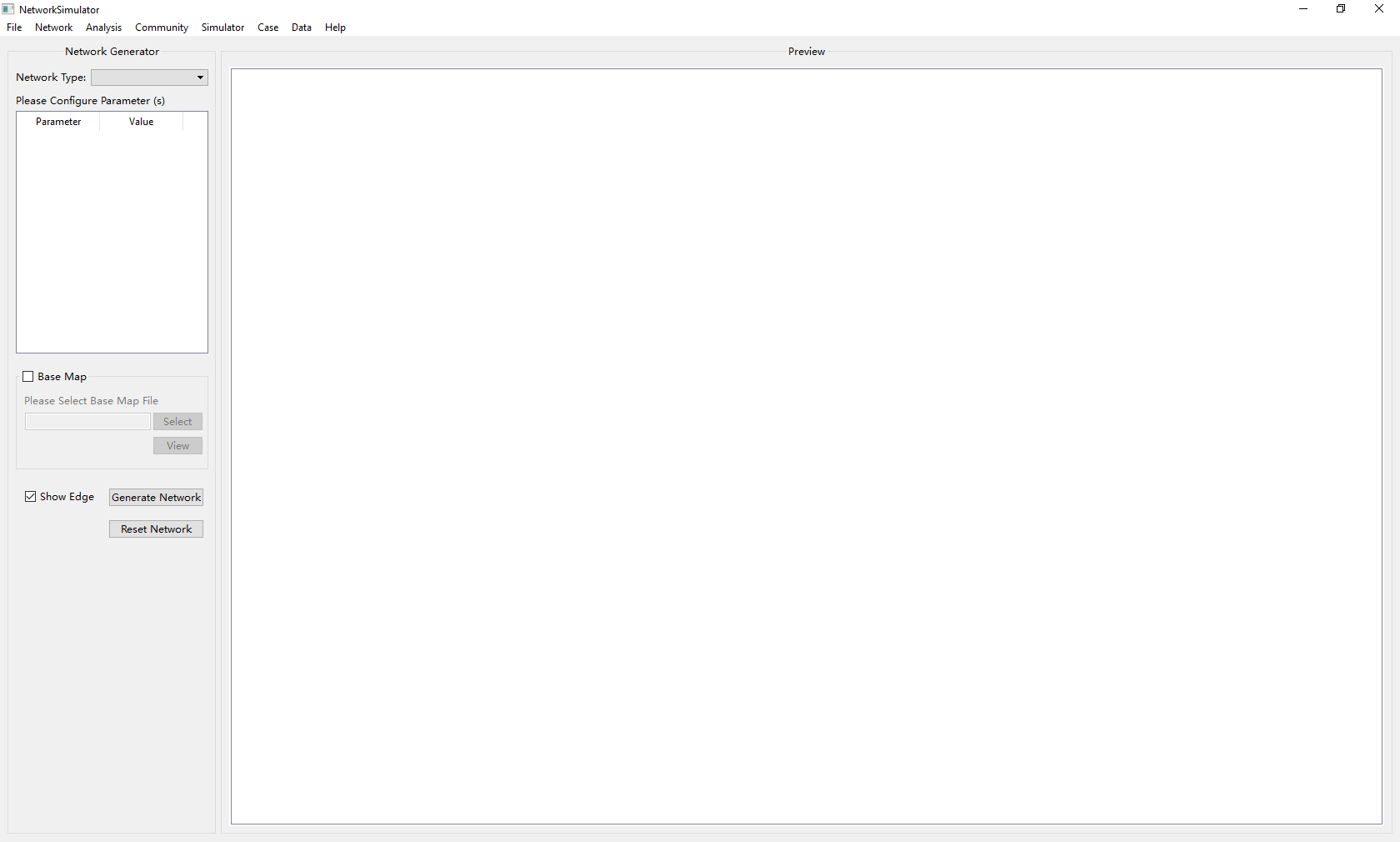
The following table is a list of menus in this software tool.

|  |  |
| --- | --- |
| Menu | Description |
| File-Exit | Exit the software |
| Network->  Generate Single Simulated network | Generate a simulated network based on a single model. |
| Network->  Generate Complex Simulated network | Generate a complex network based on one or more network model. The complex network is constructed by adding edges among several smaller networks |
| Network->Save Network | Save the current network as a txt file |
| Network->Load Network | Load a network from a txt file |
| Analysis->Network Analysis | Perform network analysis |
| Community->CNM | Conduct community detection with CNM algorithm |
| Community->GirvanNewman | Conduct community detection with GirvanNewman algorithm |
| Simulator->User Level | Simulate information diffusion on the user level |
| Simulator->City Level | Simulate information diffusion on the city level |
| Data | Provide some functions to prepare the data used in the software |

**3 NETWORK GENERATORS**

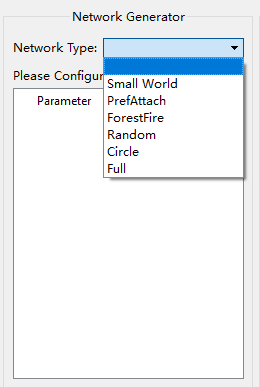
3.1 Generate a Single Simulated Network

After starting the software, the default window is for generating a single simulated network. If you are not on the interface, please choose Network→Generate Single Simulated Network to switch to the following window.

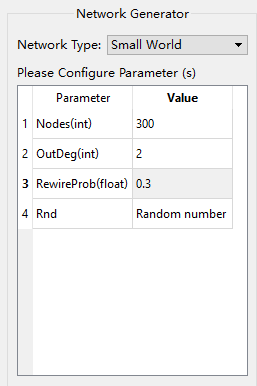
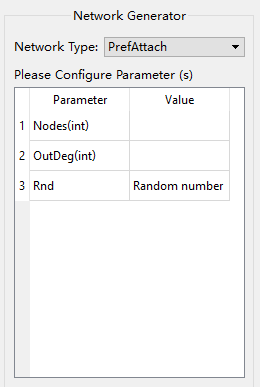


To generate a single network, please follow these steps:

(1) Select a network model from the pull-down menu.



(2) After selecting a network model, the responding parameters will load automatically. Different network model has different set of parameters. Please configure all the parameters.

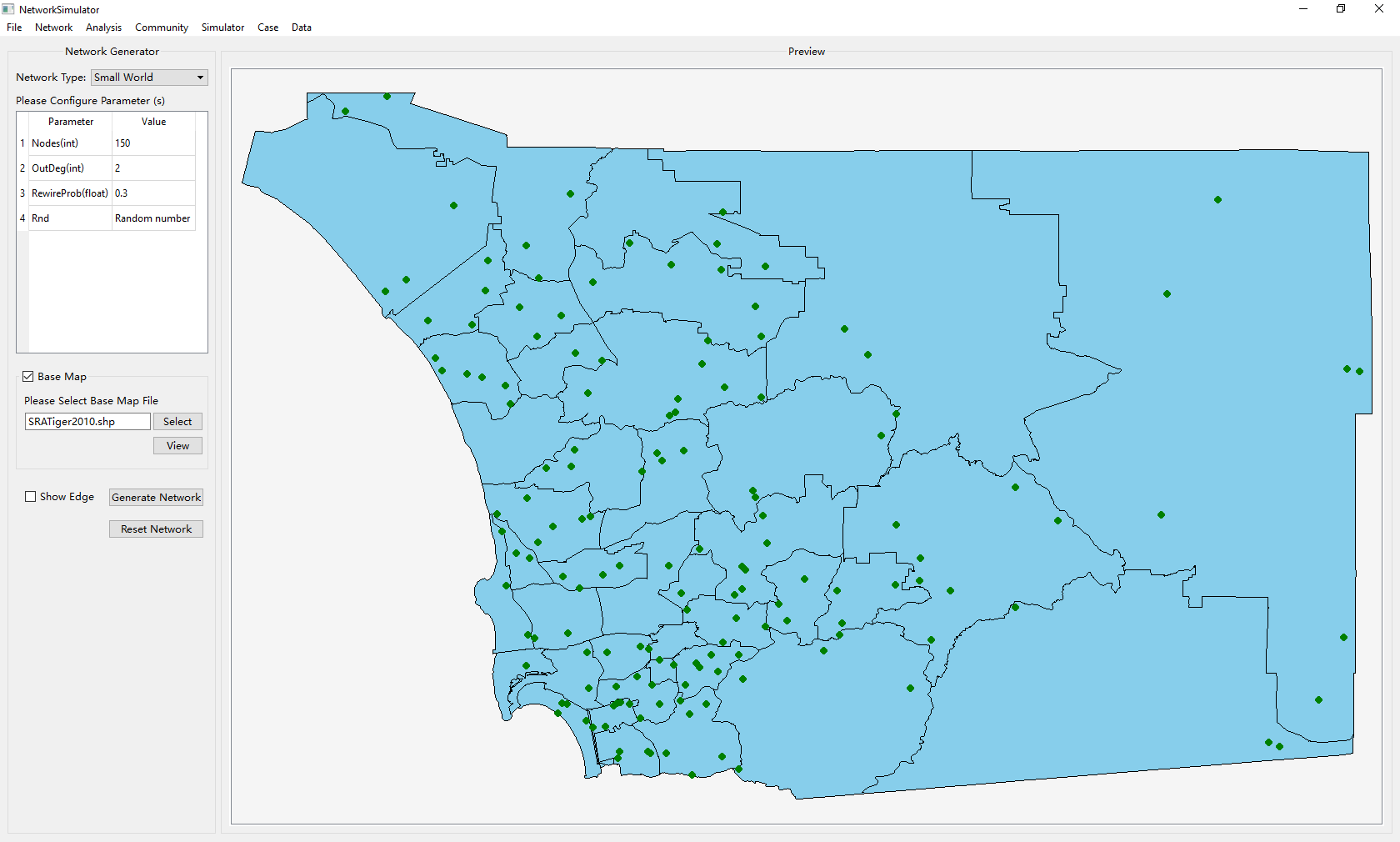
 

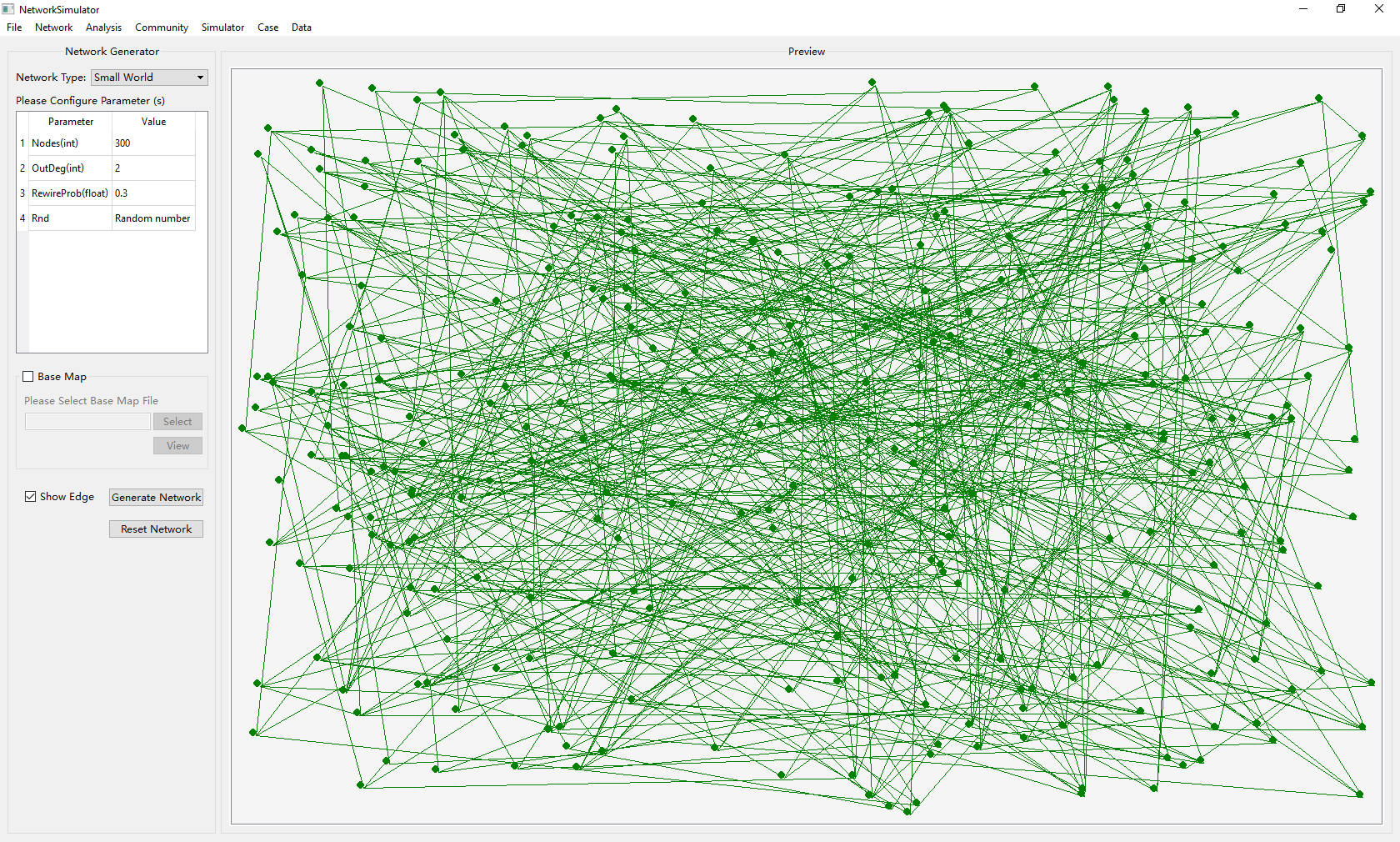
The following table lists all the parameters of each network model.

|  |  |  |
| --- | --- | --- |
| Network Model | Parameter | Description |
| Small World | Nodes | The number of nodes desired |
| OutDeg | The out degree of each node desired |
| RewireProb | Represents the probability that an edge will be rewired |
| Rnd | Random number generator (default) |
| Pref-attach Network | Nodes | The number of nodes desired |
| OutDeg | The out degree of each node desired |
| Rnd | Random number generator (default) |
| Forest Fire Network | Nodes | The number of nodes desired |
| FwdProb | Forward probability of an edge |
| BckProb | Backward probability of an edge |
| Random Network | Nodes | The number of nodes desired |
| Edges | The number of edges desired |
| Circle Network | Nodes | The number of nodes desired |
| OutDeg | The out degree of each node desired |
| Full Network | Nodes | The number of nodes desired |

(3) Check the **Base Map** to generate a network with a base map, or leave it alone to get a network without any base map. Check **Show Edge** or not to control if the network shows edges among nodes.

Here is a little bug, if you choose base map, firstly you should click **View** button to show the map, then go to the next step. Otherwise, there will be no simulated results.

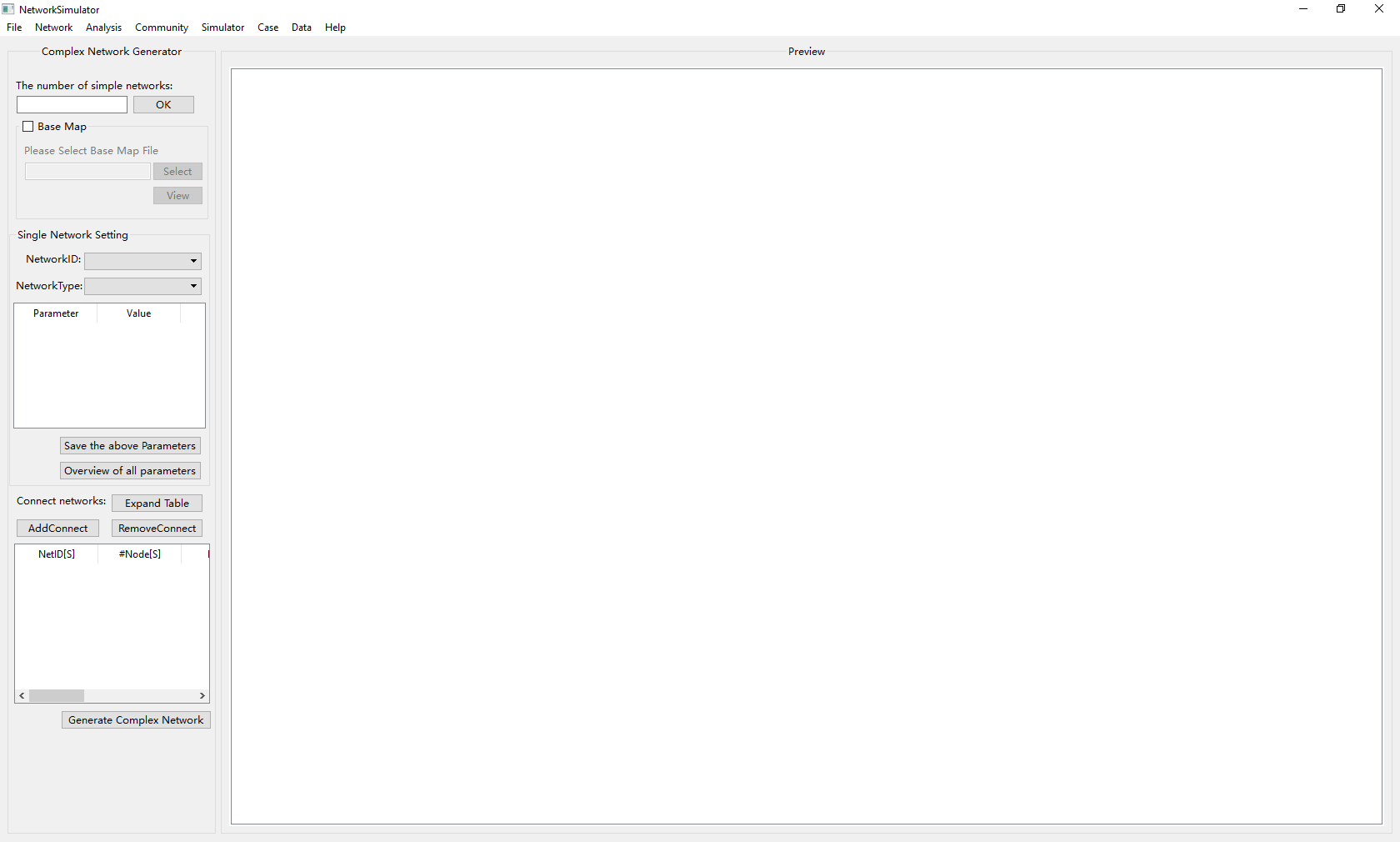




3.2 Generate a Complex Simulated Network

To generate a single network, please follow these steps:

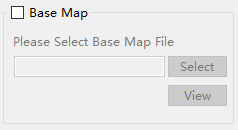
(1) Choose Network→Generate Complex Simulated Network to open the following interface.



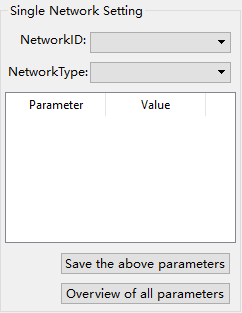
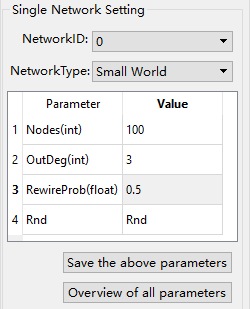
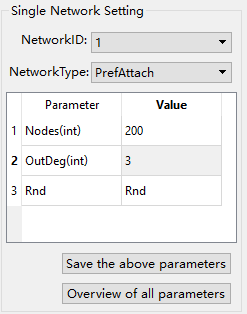
(2) Define the number of simple networks you want.

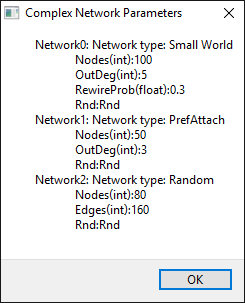


(3) Check the **Base Map** to generate a network with a base map, or leave it alone to get a network without any base map.

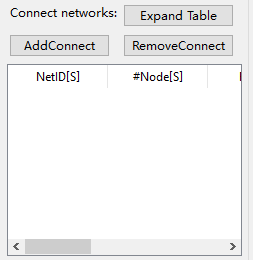


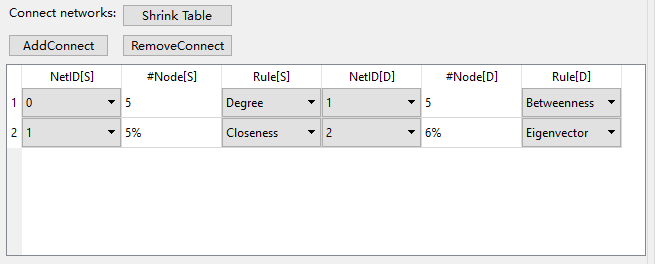
(4) Select **NetworkID** and define the corresponding parameters. You should click **Save the above parameters** to save the value of parameters for the network you choose. If you want to look at what you have defined, please click **Overview of all parameters**.

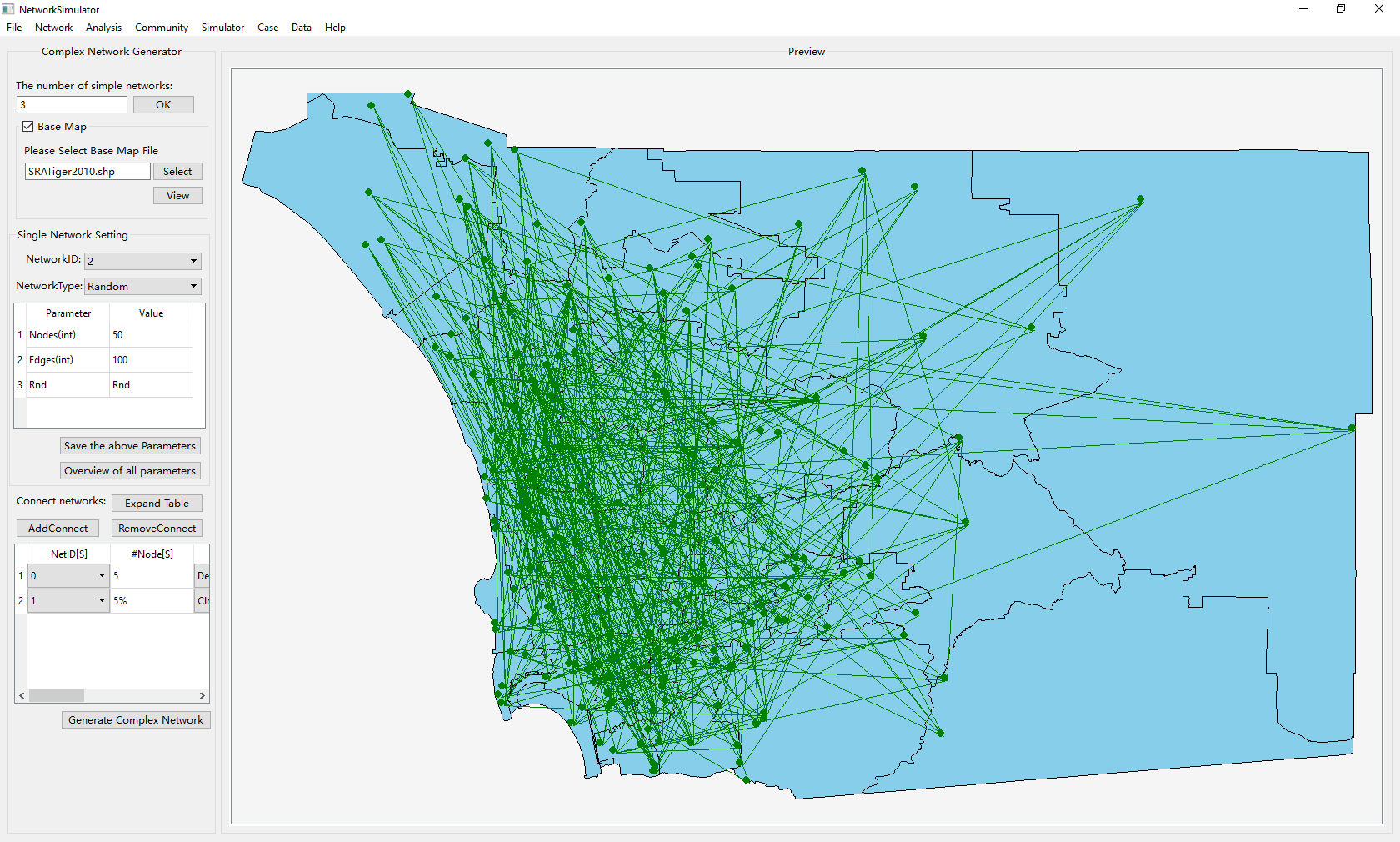


(5) Define how you want to connect these simple networks. Here you can add connections between any simple networks. For each connection, you should set network ID, the number of nodes, rule for selecting these nodes for both source network and destination network. When you set the number of nodes, you can give an integer or a percentage. In this process, you can click **Expand Table** button to expand the part block and click **Shrink Table** to restore it.





(6) After taking the above steps, click **Generate Complex Network** to generate a complex network.

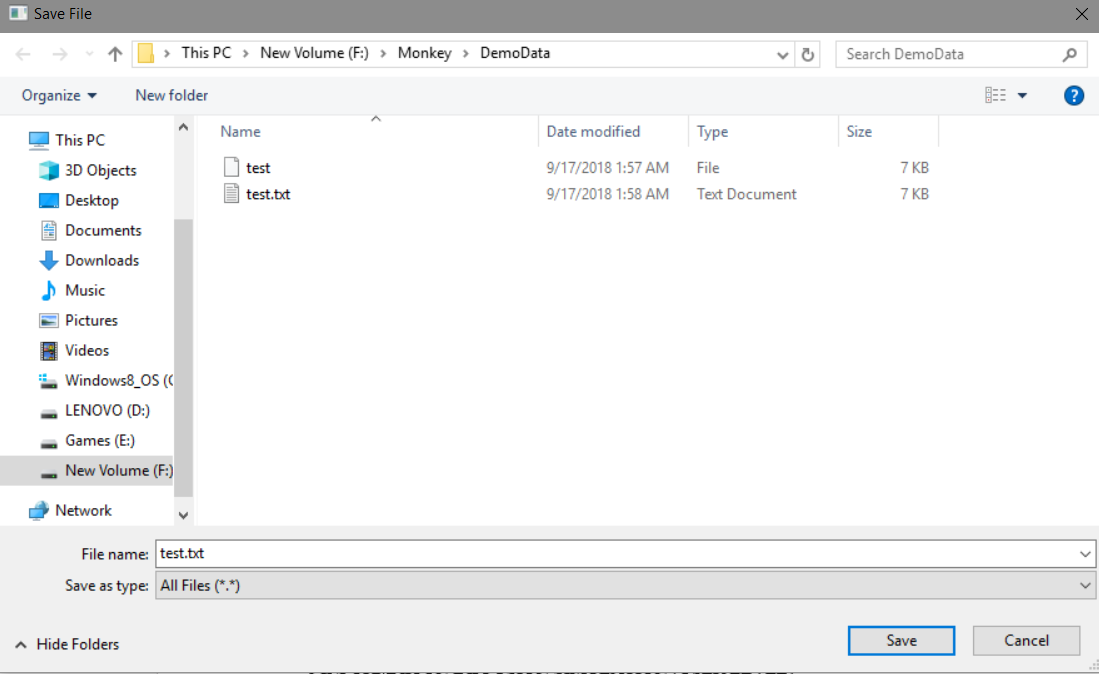


3.3 Save a Network

To save a network, please follow these steps:

1. Choose Network→Save Network to open the following interface.

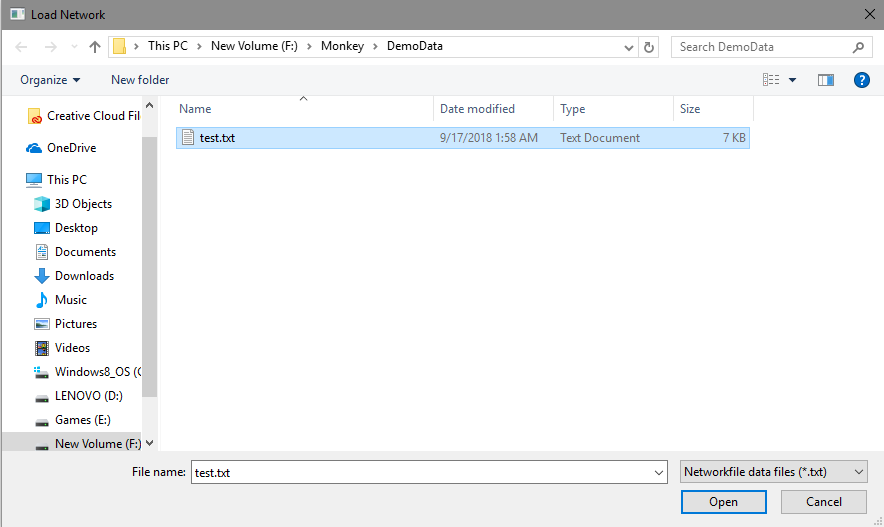
Note: File should be saved as \*.txt.

****

3.4 Load a Network

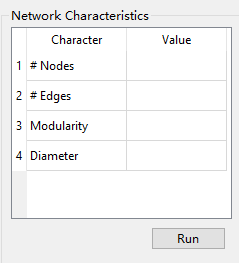
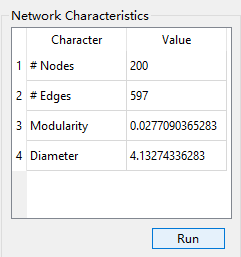
To load a network, please follow these steps:

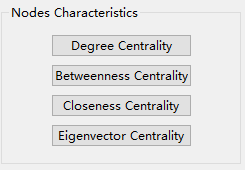
1. Choose Network→Load Network to open the following interface.

****

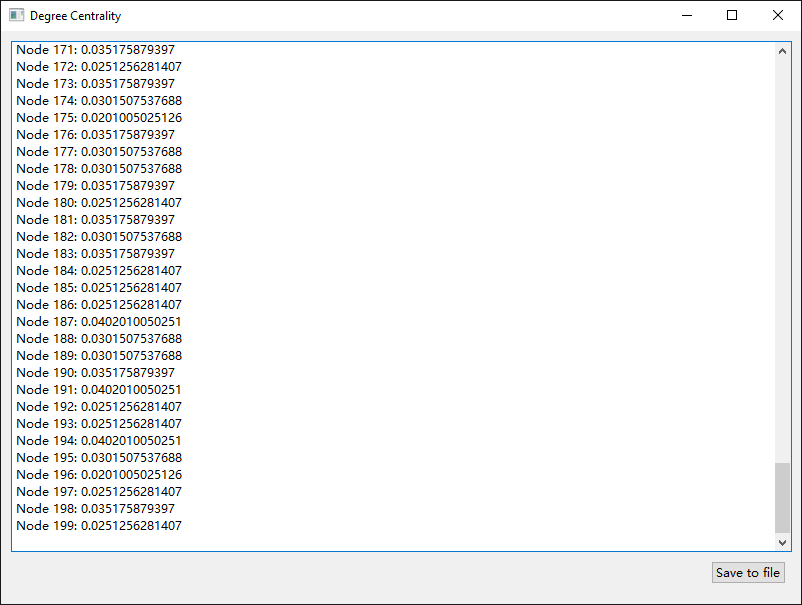
**4 NETWORK ANALYSES**

Choose **Analysis→Network** **Analysis** to switch to the interface for analyzing a network. On the left top part is a set of attributes of an entire network, please click **Run** button, you can get the value of the network’s characteristics, such as the number of nodes, the number of edges, the modularity, and the diameter.

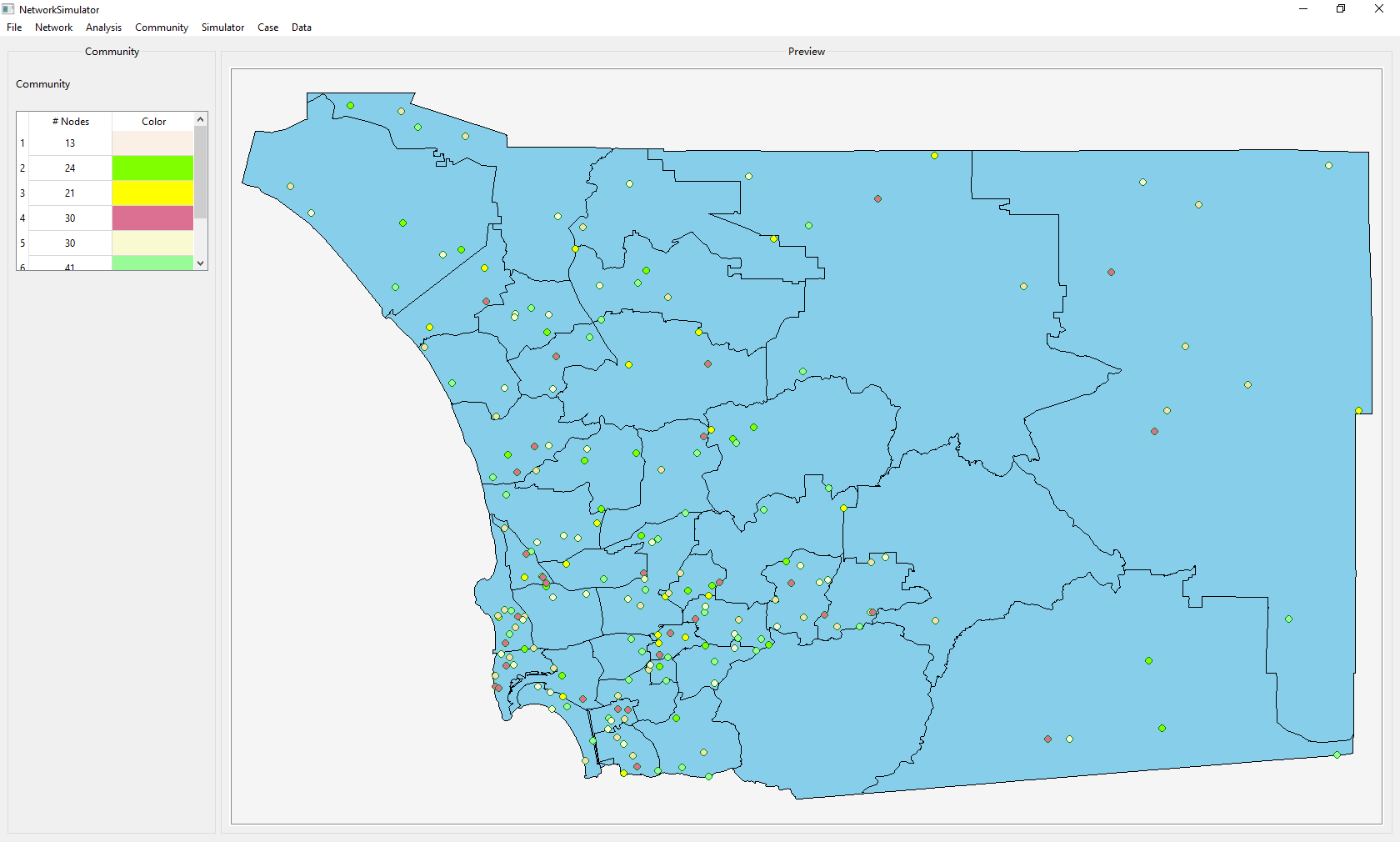
Under the characteristics of a network, there are also some characteristics for nodes, such as the degree centrality, the betweenness centrality, the closeness centrality, and the eigenvector centrality. 

Click any of the above buttons, the result will be shown in a new window. If you want to save the outcome, please click **Save to file** button which is at the corner of right bottom.



**5 COMMUNITY DETECTION**

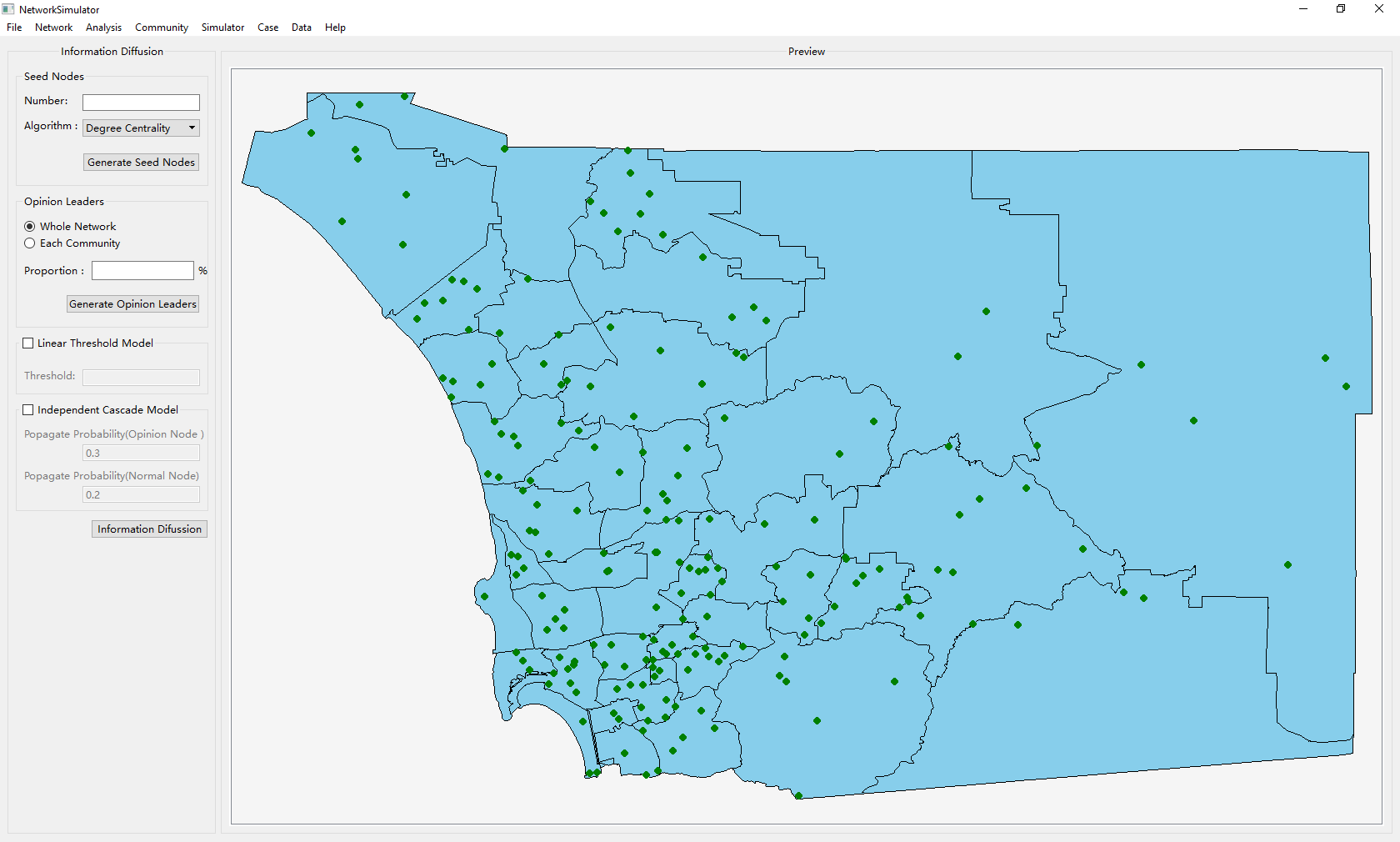
To conduct community detection, please choose **Community→CNM** or **Community→GirvanNewman**. On the left panel, there is a table to show the number of communities in the network and how many nodes there are in each community. Different color means different community and the nodes filled with same color are in a same community.

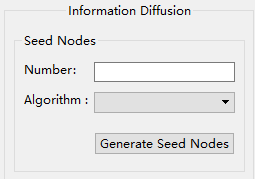


**6 INFORMAITON DIFFUSION SIMULATION (USER LEVEL)**

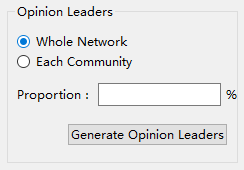
To simulate information diffusion on a city network over time, please take the following steps.

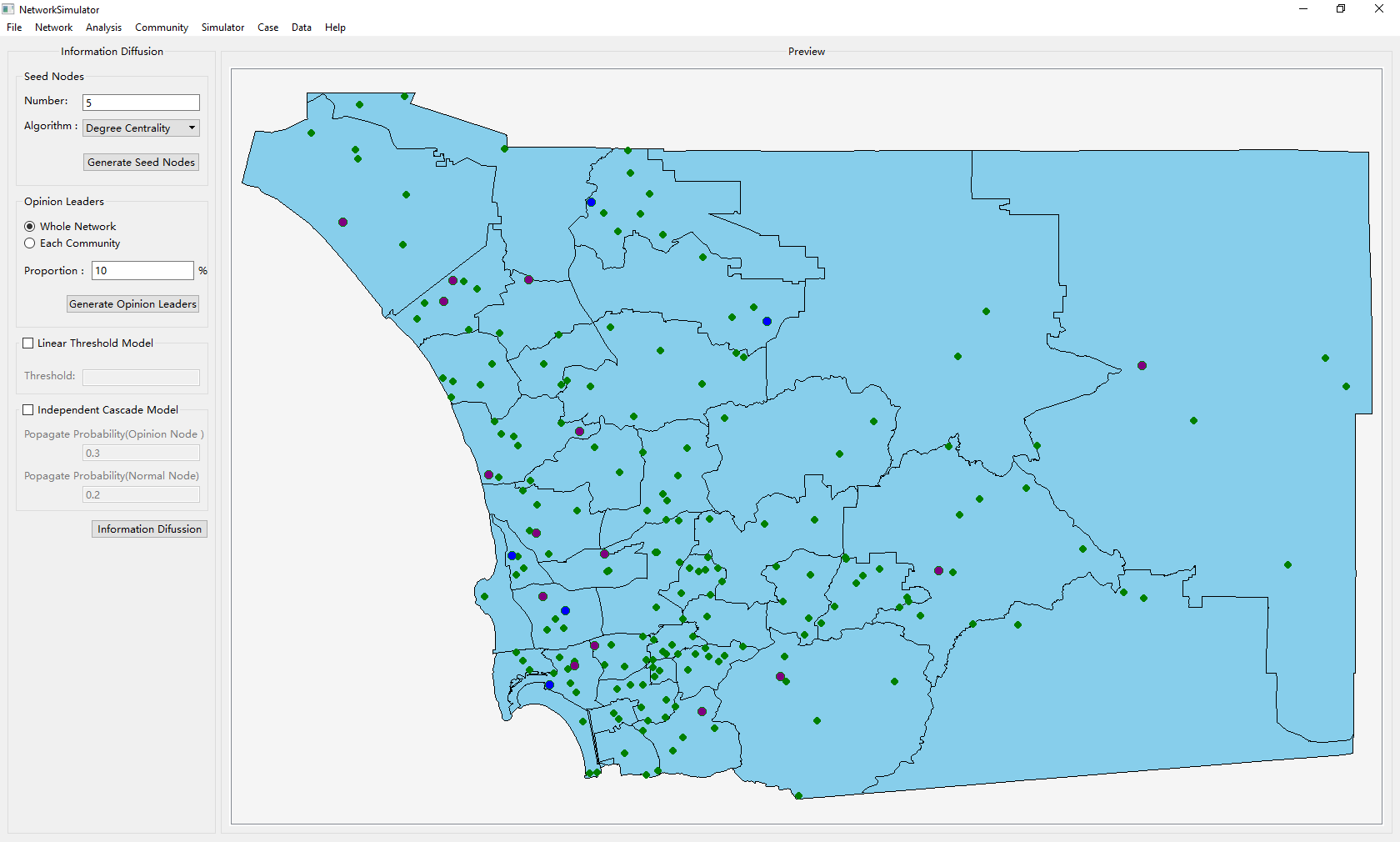
(1) Choose **Simulator→User Level** to open the following window.

 (2) Set the number of seed nodes desired and select an algorithm to choose seed nodes from all the nodes. There are five algorithms in the pull-down menu. The former four algorithms are based on a centrality to pick up the top *n* nodes to be seed nodes. The last algorithm is a randommethod which selects *n* seed nodes from all the nodes. After setting the number and the algorithm, please click **Generate Seed Nodes** button to generate seed nodes. All seed nodes will change into blue color on the **Preview** window.

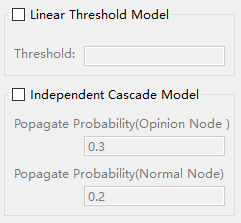


(3) Choose the option to define the percentage for generating opinion leaders. If you check **Each Community option**, please make sure you have completed the community detection and that there are communities which you are using in the network. After setting the proportion, click **Generate Opinion Leaders** to select opinion leaders. All opinion leaders will change into purple color.

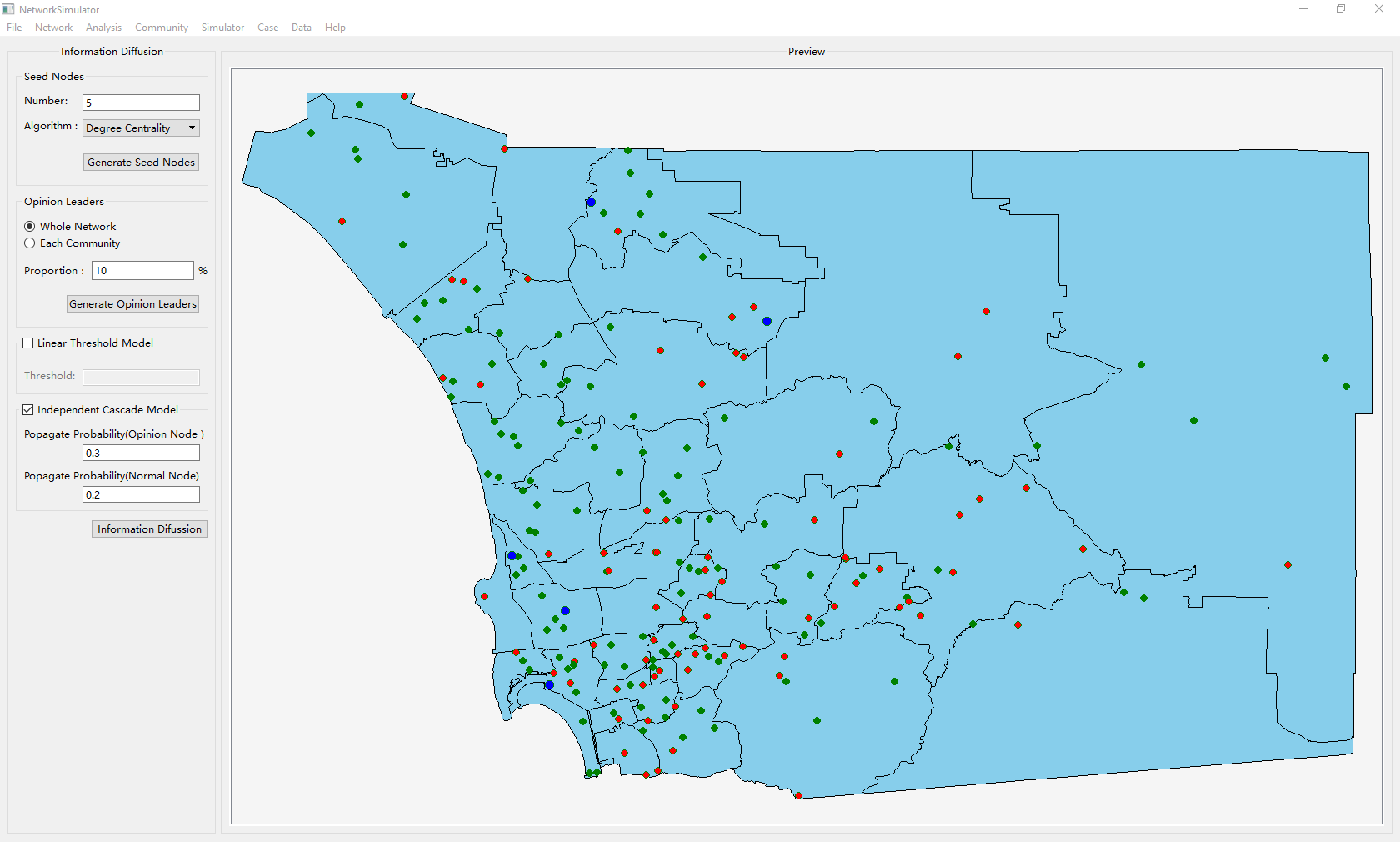




(4) Choose a model (**Linear Threshold Model** or **Independent Cascade Model**) and set the corresponding parameters for simulating the process of information diffusion. In linear threshold model, each inactive node has a uniform random threshold [0, 1] as the probability of being influenced. Each neighbor of that node can be given a weight to represent how influential it is. At each time step, the probability of successful activation depends on the total weight of its active neighbors. In independent cascade model, if one node sends a message, all nodes connected to it has equal probability to receive the message. That is to say, an independent cascade model is sender-centric while a linear threshold model is receiver-centric. If you check **Linear Threshold Model**, please define **Threshold**. Otherwise, when you check **Independent Cascade Model**, please set **Propagate Probability (Seed Node)** and **Propagate Probability (Normal Node)**.



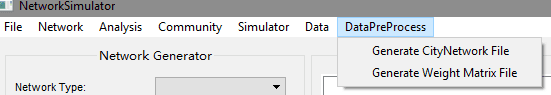
(5) After taking the above steps, click **Information** **Diffusion** button to start the simulation. You will see the dynamic process on the right part of the window. When the process is finished, all inactive nodes will be shown and filled in green color.



**7 INFORMAITON DIFFUSION SIMULATION (CITY LEVEL)**

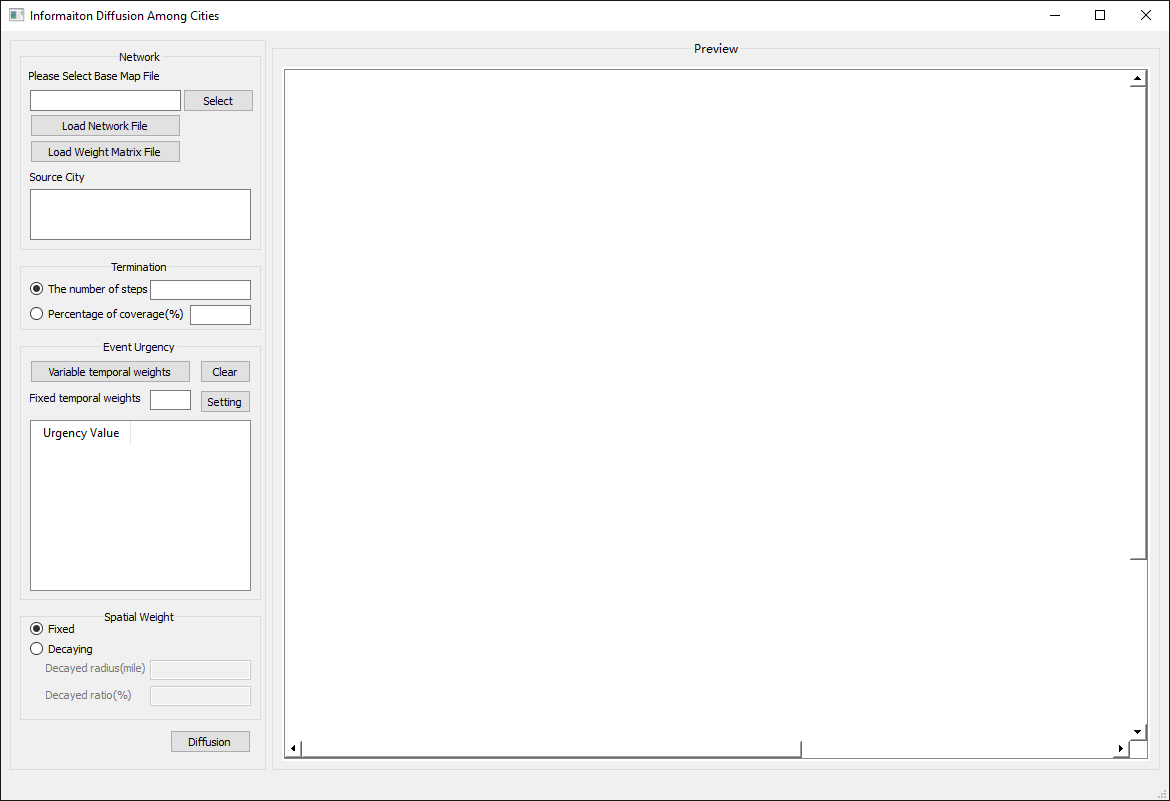
To simulate information diffusion on a city network over time, please switch to the main window.

(1) Choose **DataPreProcess→Generate CityNetwork File** to generate citynetwork.txt  
 Please choose a valid city location file which contains XY city coordinates.

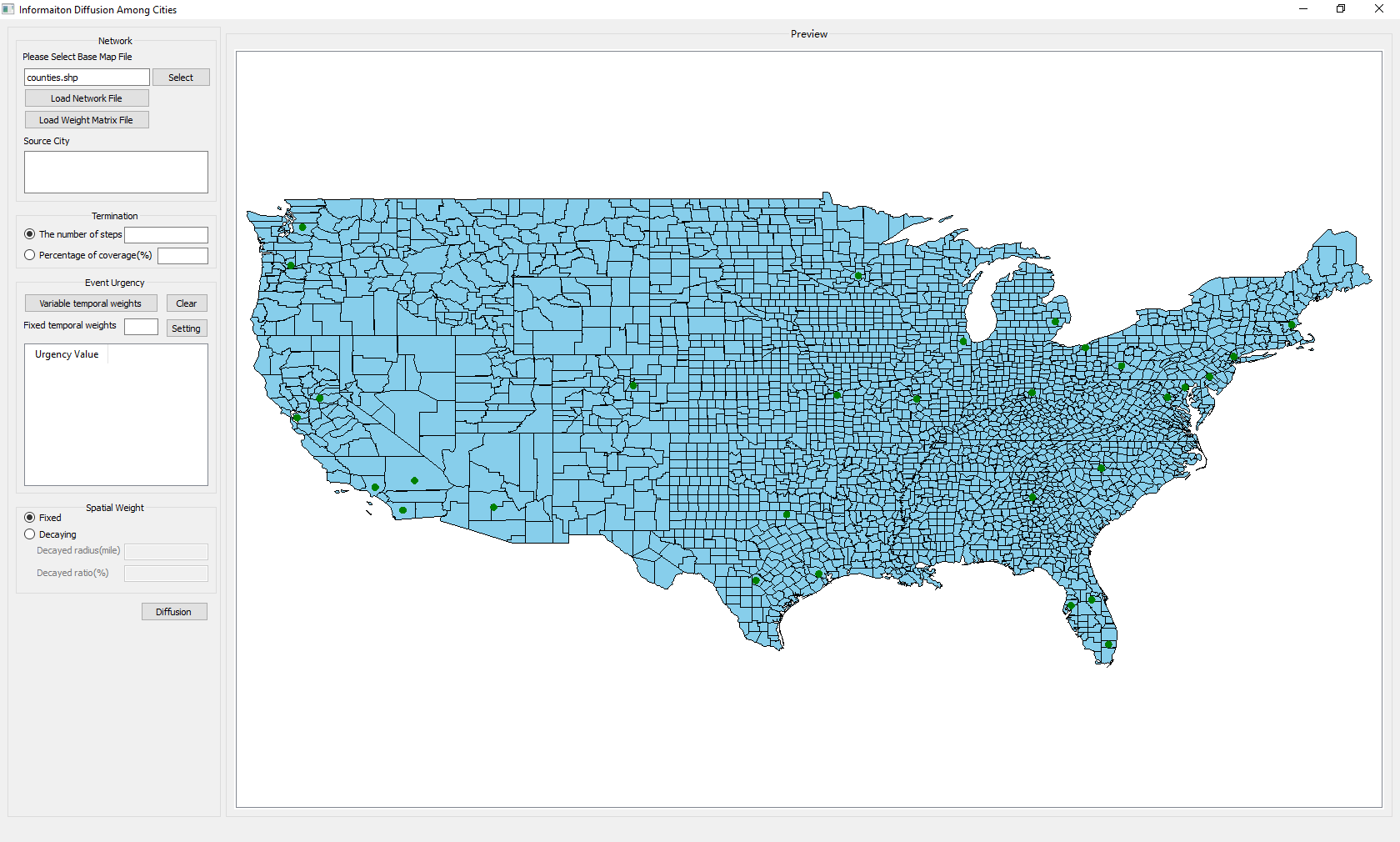


(2) Choose **DataPreProcess→Generate Weight Matrix File** to generate Weight Matrix File

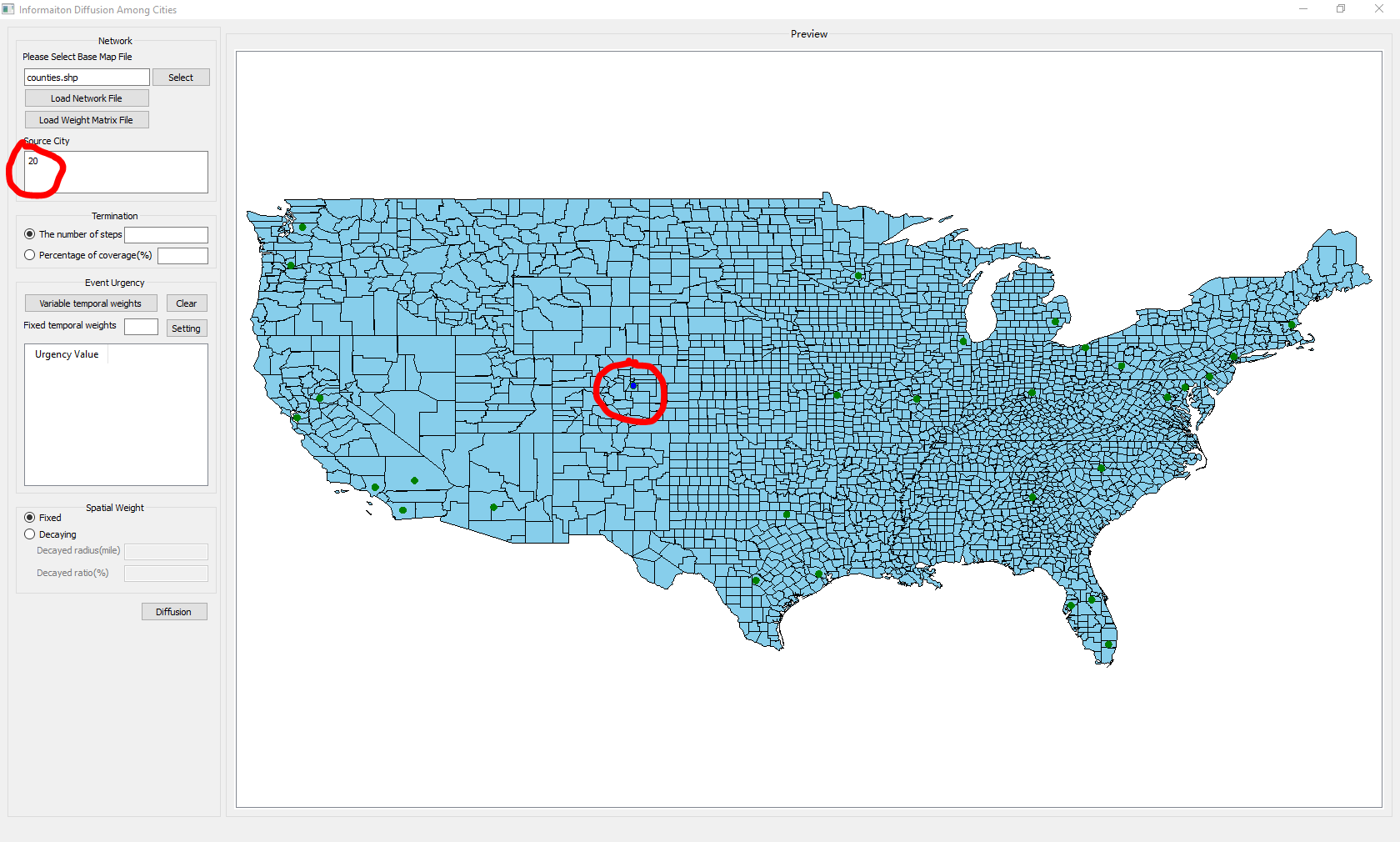
(3) Choose **Simulator→City Level** to open the following window.7



(4) Click **Select** button on the left panel to select a base map. Click **Load Network File** button load city network. Click **Load Weight File** to apply a weight matrix to the loaded city network.

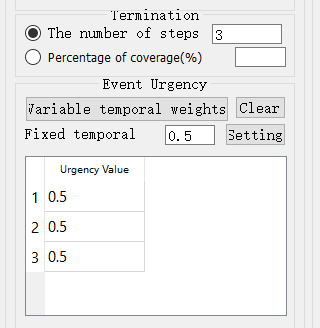


(5) Input one or more source cities at the source city textbox. You can also choose a source city through double clicking a node on the map.

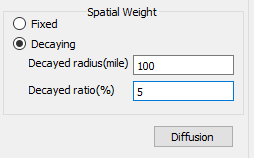


(6) Set the way to terminate a simulation. Check **The number of steps** and input a positive integer, or check **Percentage of coverage** and input an integer between 0 and 100.

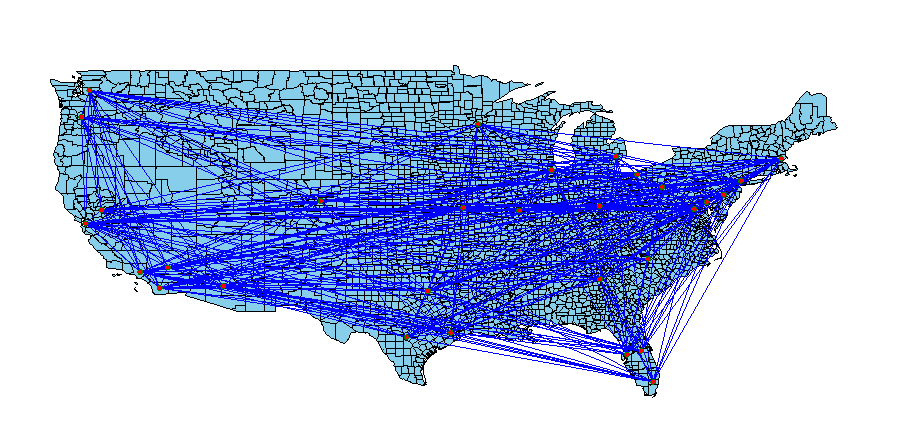
(7) Set the event urgency. If you checked **The number of steps in the step (4)**, here you can set each event urgency for the responding step. You can click **Variable temporal weights** to set each event urgency manually. If you want to set an equal event urgency for all the steps, please input the weight in the textbox after **Fixed temporal** button. Then click **Setting** to set the weights. Of course, you can still change any of these weights to let some steps have different event urgencies. If you check **Percentage of coverage** in the step (4), now you just can set an event urgency for the entire simulation. If you want to clear what you have set, please click **Clear**.

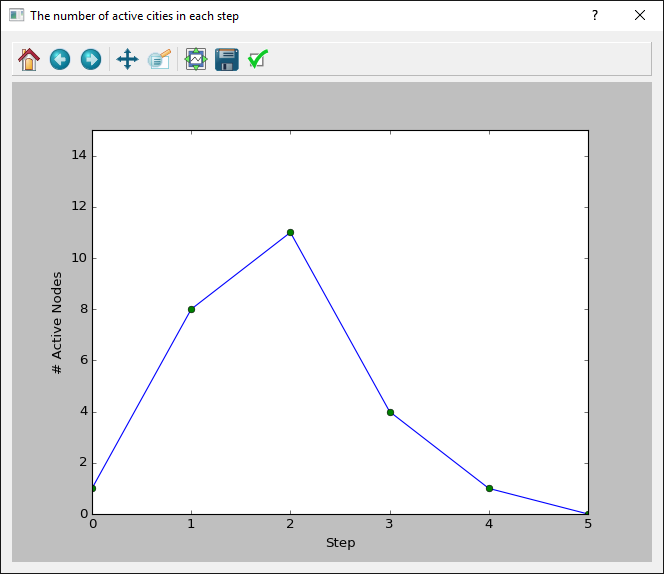
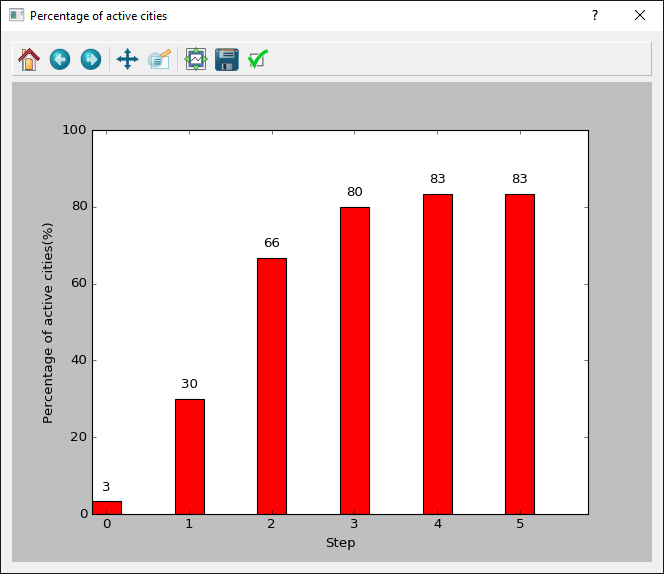


(8) Set the spatial weight. You can choose **Fixed** to set a fixed weight to all the cities no matter the distance among them. If you hope that the distance affects the information diffusion, please check **Decaying**, and set Decayed radius (miles) and Decayed ratio (%) respectively.



(9) After taking the aforementioned steps, click **Diffusion** button to start the simulation. On the diffusion window, you will see the process of message propagation among cities step by step. The red nodes denote the cities that have received the message. The red line means current diffusion and the blue lines are past diffusion edges. Upon the diffusion is finished, the following windows will popup. One is the number of active nodes at each step, and the other is the accumulative percentage of coverage at every step. If there are green nodes on the map, which means these nodes were inactive.

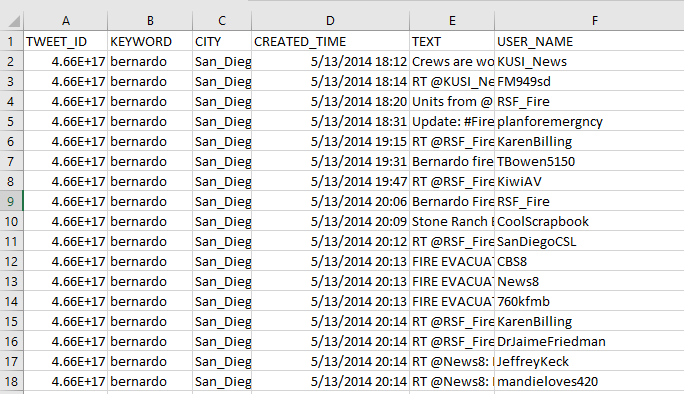


**8 DATA PROCESSING**

8.1 Extract a retweet network

Load a csv file that at least includes these columns, USER\_NAME, CREATE\_TIME, TEXT, as follows.



USER\_NAME: the user who post or retweet a tweet.

CREATE\_TIME: the time when a tweet was post.

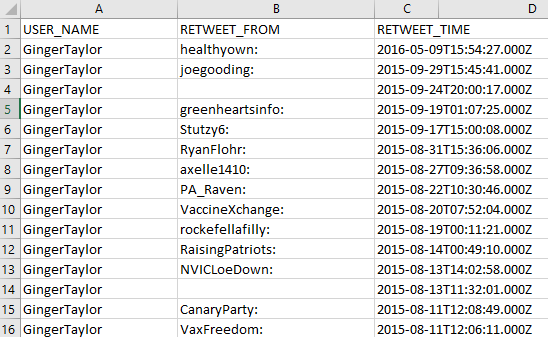
TEXT: the content of a tweet.

The output is also a csv file which includes ["USER\_NAME", "ADOPTED\_ FROM", "ADOPTED\_ TIME"]. Here is an example:

USER\_NAME: a user who posted a tweet.

RETWEET\_FROM: the user which a tweet retweet from. Any record where RETWEET\_FROM is empty means the tweet is not retweet from other user.

RETWEET\_TIME: the time when a tweet was post.



8.2 Extract a mention network

The format of an input file is same as the function for exacting a retweet network. The result is similar to the output of a retweet network. The differences between are the second column and the third column, RETWEET\_FROM and RETWEET\_TIME are replaced with MENTIONED\_USERS and MENTIONED\_TIME.

8.3 Compute a weight matrix among cities

To generate a weight matrix for expressing the relationship among cities, you need three files that are about cities, users, and tweets separately.

The following table presents what columns each file should include.

|  |  |  |
| --- | --- | --- |
| File | | Columns |
| City | CITY\_ID, CITY\_NAME | |
| Tweets | USER\_NAME, CREATE\_TIME, TEXT | |
| User | USER\_NAME, CITYID | |

The output is a txt file which include a matrix. Each row is a set of weights that means a city retweets message from other cities. The row is sorted by CITY\_ID in the city file.